

Docket No.: 58799-095

**PATENT**

**IN THE UNITED STATES PATENT AND TRADEMARK OFFICE**

In re Application of	:	Customer Number: 20277
	:	
Shougo SHIMIZU, et al.	:	Confirmation Number:
	:	
Serial No.:	:	Group Art Unit:
	:	
Filed: September 15, 2003	:	Examiner:
	:	
For: SYSTEM OF GENERATING PROCEDURE FOR DIGITAL SIGNATURE AND ENCRYPTION TO XML		

**CLAIM OF PRIORITY AND  
TRANSMITTAL OF CERTIFIED PRIORITY DOCUMENT**

Mail Stop  
Commissioner for Patents  
P.O. Box 1450  
Alexandria, VA 22313-1450

Sir:

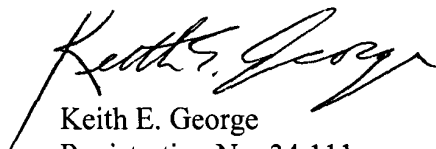
In accordance with the provisions of 35 U.S.C. 119, Applicants hereby claim the priority of:

**Japanese Patent Application No. 2003-046683, filed February 25, 2003**

A Certified copy is submitted herewith.

Respectfully submitted,

MCDERMOTT, WILL & EMERY



Keith E. George  
Registration No. 34,111

600 13<sup>th</sup> Street, N.W.  
Washington, DC 20005-3096  
(202) 756-8000 KEG:prg  
Facsimile: (202) 756-8087  
**Date: September 15, 2003**

日 本 国 特 許 庁  
JAPAN PATENT OFFICE

58799-095  
Shimizu et al.  
Sept. 15, 2003

MaDermott, Will & Emery

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office

出 願 年 月 日  
Date of Application:

2003年 2月25日

出 願 番 号  
Application Number:

特願2003-046683

[ ST.10/C ]:

[ JP 2003-046683 ]

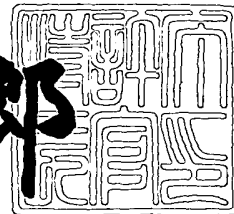
出 願 人  
Applicant(s):

株式会社日立製作所

2003年 5月13日

特 許 庁 長 官  
Commissioner,  
Japan Patent Office

太田 信一郎



出証番号 出証特2003-3035594

【書類名】 特許願

【整理番号】 K02012211A

【あて先】 特許庁長官殿

【国際特許分類】 H04L 9/00

【発明者】

【住所又は居所】 神奈川県川崎市麻生区王禅寺 1 0 9 9 番地 株式会社日立製作所システム開発研究所内

【氏名】 清水 將吾

【発明者】

【住所又は居所】 神奈川県横浜市戸塚区戸塚町 5 0 3 0 番地 株式会社日立製作所ソフトウェア事業部内

【氏名】 砂田 英昭

【発明者】

【住所又は居所】 神奈川県横浜市戸塚区戸塚町 5 0 3 0 番地 株式会社日立製作所ソフトウェア事業部内

【氏名】 清水 英則

【発明者】

【住所又は居所】 神奈川県川崎市麻生区王禅寺 1 0 9 9 番地 株式会社日立製作所システム開発研究所内

【氏名】 秋藤 俊介

【発明者】

【住所又は居所】 神奈川県川崎市麻生区王禅寺 1 0 9 9 番地 株式会社日立製作所システム開発研究所内

【氏名】 工藤 裕

【特許出願人】

【識別番号】 000005108

【氏名又は名称】 株式会社 日立製作所

【代理人】

【識別番号】 100075096

【弁理士】

【氏名又は名称】 作田 康夫

【手数料の表示】

【予納台帳番号】 013088

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 XML署名・暗号化手順生成システム

【特許請求の範囲】

【請求項1】

XML文書に対する署名・暗号化手順を生成するシステムであって、  
利用するそれぞれのWebサービスからXML署名およびXML暗号化の手順  
を記載した手順書を取得する手段と、

前記手順書からXML署名およびXML暗号化の対象となる要素のスキーマを  
取得する手段と、

取得した手順書とスキーマを解析し、それらの要求を満たす適切なXML署名  
およびXML暗号化手順を出力する手段とを備えたことを特徴とする署名・暗号  
化手順生成システム。

【請求項2】

前記XML署名およびXML暗号化手順に従ってXML署名およびXML暗号  
化を行うプログラムを自動生成することを特徴とする請求項1の署名・暗号化手  
順生成システム。

【請求項3】

Webサービスにおけるメッセージ送信時に、該メッセージに対して、生成さ  
れたXML署名およびXML暗号化プログラムを実行し、その結果を送信するこ  
とを特徴とする請求項2の署名・暗号化手順生成システム。

【請求項4】

前記XML署名およびXML暗号化プログラムの生成時に、XML署名および  
XML暗号化を行うXML要素のXMLスキーマの識別子と前記XML署名およ  
びXML暗号化手順書のリストの識別子と前記XML署名およびXML暗号化プ  
ログラムとの対応を記憶装置に格納することを特徴とする請求項2の署名・暗号  
化手順生成システム。

【請求項5】

XML文書の送信時に、XMLスキーマの識別子とXML署名およびXML暗  
号化手順書のリストの識別子から、前記対応を参照してXML署名およびXML

暗号化モジュールを決定し、前記XML文書に対して前記XML署名およびXML暗号化プログラムを実行し、その結果を送信することを特徴とする請求項4の署名・暗号化手順生成システム。

【請求項6】

XML文書の送信時に、前記XML文書に記述されたWebサービスの識別子からXML署名およびXML暗号化の手順書を取得し、前記XML署名およびXML暗号化の手順書からXML署名およびXML暗号化プログラムを生成し、前記XML文書に対して前記XML署名およびXML暗号化プログラムを実行し、その結果を送信することを特徴とする請求項2記載の署名・暗号化手順生成システム。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、XML署名およびXML暗号化を行う方法に関するものである。

【0002】

【従来の技術】

電子署名または暗号化を行うシステムの開発において、従来の手法では、上流工程によりあらかじめ設計された署名または暗号化手順に基づき、開発者が手作業で署名または暗号化を行うアプリケーションと署名の検証または復号化を行うアプリケーションの開発を行っている。

【0003】

また、XML文書に対する署名および暗号化の方法として、XML署名およびXML暗号化と呼ばれる技術がW3Cという標準化団体により標準化されている。これらの技術では、XMLの部分文書に対する署名および暗号化が可能である。また、同一XML文書内で複数の署名データおよび暗号化データを表現することができる。なお、XML文書の署名に関する公知文献として、例えば特開平6-224896号公報(特許文献1)には回覧途中で文書内容の変更を可能にしたデジタル署名の生成方法と電子化文書の認証方法が示される。

【0004】

【特許文献1】特開平6-224896号公報

【0005】

【発明が解決しようとする課題】

Webサービスは各サービスが独立して設計されるため、既存のWebサービスを利用するようなWebサービスの開発においては、利用するWebサービスが要求する署名または暗号化の手順に従って署名および暗号化を行う必要がある。特に、複数のWebサービスを利用する場合、各々のWebサービスから独立に要求される署名および暗号化手順をすべて満たすように、開発者が手作業で署名および暗号化手順を調べ、その手順に従って署名および暗号化を行うようなプログラムを開発する必要がある。例えば、親子関係にある二つの要素parent、childへのXML暗号化がそれぞれWebサービスA、Bから要求されており、かつAに対してはchild要素の内容が見えてはならないという要求がある場合、まずBに対してchild要素をXML暗号化し、次にAに対してparent要素を暗号化する必要がある。

【0006】

本発明の目的は、このように複数のWebサービスから要求されるXML署名およびXML暗号化の手順を自動的に解析し、すべての要求を満たすような手順でXML署名およびXML暗号化を行うようなプログラムを自動生成することにより、Webサービス開発者の負担を軽減することにある。

【0007】

【課題を解決するための手段】

上記の課題を解決するために、本発明では、以下のステップによりXML署名およびXML暗号化を行う。

(1) 利用するそれぞれのWebサービスからXML署名およびXML暗号化の手順を記載した手順書を取得し、XML署名およびXML暗号化の対象となる要素のスキーマを取得する。

(2) 上記ステップにより取得した手順書とスキーマを解析し、すべての要求を満たすための適切なXML署名およびXML暗号化手順を出力する。

(3) 上記ステップにより出力された手順に従ってXML署名およびXML暗号

化を行うようなプログラムを自動生成する。

(4) W e b サービスにおけるメッセージ送信時に、該メッセージに対して、上記ステップにより生成されたXML署名およびXML暗号化プログラムを実行し、その結果を送信する。

【0008】

上記のようにあらかじめXML署名およびXML暗号化プログラムを生成する代わりに、実行時にXML署名およびXML暗号化手順の解析とXML署名およびXML暗号化プログラムの生成を行ってもよい。この場合、以下のステップによりXML署名およびXML暗号化を行う。

(1) W e b サービスにおけるメッセージ送信時に、メッセージ内の記述からXML署名およびXML暗号化手順書のURIとXML署名およびXML暗号化の対象となる要素のスキーマのURIを特定し、該手順書と該スキーマを取得する。

(2) 上記ステップにより取得した手順書とスキーマを解析し、すべての要求を満たすための適切なXML署名およびXML暗号化手順を出力する。

(3) 上記ステップにより出力された手順に従ってXML署名およびXML暗号化を行うようなプログラムを自動生成する。

(4) 該メッセージに対して、上記ステップにより生成されたXML署名およびXML暗号化プログラムを実行し、その結果を送信する。

【0009】

【発明の実施の形態】

以下、図1から図17を用いて、本発明の一実施形態によるXML署名およびXML暗号化方法について説明する。

【0010】

まず、図1を用いて、本実施形態によるXML署名およびXML暗号化方法の概要について説明する。各構成部分の詳細については後に説明する。以下では、ユーザが既存のW e b サービスを利用するようなW e b サービスを新規に開発する場合を考える。まず、ユーザはW e b サービス呼出し順定義画面102を利用して、新規に開発するW e b サービスが利用するW e b サービスおよびそれらの



間の呼出し順を定義する。102はWebサービスの呼出し順や各Webサービスが要求するXML署名・暗号化手順書のURIを記述したデータ103を生成する。

XML署名・暗号化モジュール生成部101はXML署名・暗号化を行うモジュールを生成する処理部分であり、XML署名・暗号化手順書取得部105、XML署名・暗号化手順解析部107、XML署名・暗号化モジュール出力部110、XML署名・暗号化モジュール登録部112からなる。

#### 【0011】

101は103を入力として呼び出され、まず105が実行される。105は103を解析し、103に記述されている各WebサービスのXML署名・暗号化手順書URIからXML署名・暗号化手順書104を取得し、それらをXML署名・暗号化手順書リスト106として出力する。

次に、107が実行される。107は106とXML署名・暗号化の対象となる要素のXMLスキーマ108を読み込み、106のすべての要求手順を満たすためのXML署名・暗号化手順を解析し、解析の結果決定された手順をXML署名・暗号化手順109として出力する。

次に、110が実行される。110は109を解析し、108のスキーマに適合するXML文書に対して109に記述された手順でXML署名・暗号化を行うようなXML署名・暗号化モジュール111を生成する。

次に、112が実行される。112は108のXMLスキーマと103のWebサービス呼出し順と111のXML署名・暗号化モジュールとの対応をXML署名・暗号化モジュール対応表113に登録する。

XML署名・暗号化モジュールの実行時は以下のように動作する。XML署名・暗号化実行部114は108のスキーマに適合するXML文書115を受け取り、115に記述されたXMLスキーマおよびWebサービス呼出し順の情報と113から対応するXML署名・暗号化モジュールを決定し、115に対して該モジュールを実行する。その結果得られたXML文書116をWebサービスの送信文書として送信する。

#### 【0012】

次に、図 2 を用いて、システム全体のハードウェア構成について説明する。205 の外部記憶装置 1 はプログラムが格納されている外部記憶装置であり、101、105、107、110、112、114 が格納されている。206 の外部記憶装置 2 はデータが格納されている外部記憶装置であり、103、104、106、108、109、111、113、115、116 が格納されている。203 は中央処理演算装置、204 は主記憶装置である。205 に格納されているプログラムが呼び出されたとき、該プログラムの内容が 204 に読み込まれ、203 により処理される。該プログラムが 206 に格納されているデータを必要としたとき、該データの内容が 204 に読み込まれ、該プログラムにより処理される。また、該プログラムが 204 のデータを外部記憶装置に出力するとき、該データの内容が 206 に書き込まれる。201 は CRT ディスプレイ等の表示装置、202 はキーボードやマウス等の入力装置である。

#### 【0013】

図 3 は Web サービス呼出し順定義画面 102 の例である。301 は開発するシステムの画面および画面間の遷移を定義する画面遷移定義画面の例である。ここで、302～304 は画面を表し、305、306 は画面間の遷移を表す。Web サービス呼出し順定義画面 102 は呼び出す Web サービスの名称やそれらの間の呼出し順を定義する画面である。ここで、307～309 は Web サービスを表し、310、311 はそれらの間の呼出し順を表す。312 は 306 の画面遷移の間に呼び出される Web サービスが 307 のチケット手配サービス、308 のホテル予約サービス、309 のカード決済サービスをこの順で呼び出すことを意味する。

#### 【0014】

図 4 は 102 により定義された Web サービス呼出し順 103 の例である。406～408 の各行はそれぞれ 307～309 に対して定義された情報に対応する。ここで、ID401 はツール内で各 Web サービスを一意に特定するための識別子である。次 ID402 は次に呼び出す Web サービスの識別子であり、401 のいずれかの値を参照する。例えば、406 において次 ID がホテル予約サービスの ID である B であることは、310 に示される Web サービス呼出し順

を表現する。名称 4 0 3 は W e b サービスの名称、W e b サービス U R I 4 0 4 は該 W e b サービスをインターネット上で一意に特定するための識別子である。XML 署名・暗号化手順書 U R I 4 0 5 は該 W e b サービスが要求する XML 署名・暗号化手順が記述された XML 署名・暗号化手順書をインターネット上で一意に特定するための識別子である。

## 【 0 0 1 5 】

図 5 は XML 署名・暗号化手順書 1 0 4 の例である。5 0 1 ~ 5 0 3 はそれぞれ 4 0 6 ~ 4 0 8 の XML 署名・暗号化手順書 U R I 4 0 5 に記述されていると仮定する。5 0 1 の意味は以下の通りである。まず、順序 5 0 4 が 1 である `t i c k e t s` 要素の暗号化を A E S アルゴリズムにより行う ( 5 0 8 ) 。次に、順序が 2 である `u s e r i n f o` 要素の暗号化を D E S e d e アルゴリズムにより行う ( 5 0 9 ) 。最後に、順序が 3 である `r o o t` 要素に対する署名を D S S アルゴリズムにより行う ( 5 1 0 ) 。すなわち、チケット手配サービスを利用する W e b サービスはこの手順に従って XML 署名および暗号化を行った XML 文書を送信する必要がある。5 0 2、5 0 3 に対しても同様である。

## 【 0 0 1 6 】

図 6 は XML 署名・暗号化手順書取得部 1 0 5 のフローチャート例である。以下、本フローチャートに従って XML 署名・暗号化手順書取得時の動作を説明する。図 4 に示した 1 0 3 の各行について、以下の 6 0 2 から 6 0 4 のステップを行う ( 6 0 1 ) 。まず、1 0 3 の XML 署名・暗号化手順書 U R I 4 0 5 で示された U R I から XML 署名・暗号化手順書を取得する ( 6 0 2 ) 。次に、6 0 2 で取得した手順書の各行 `t` に対して、次の 6 0 4 のステップを行う ( 6 0 3 ) 。6 0 4 では、`t` に該 W e b サービスの I D を追加し、`t` を XML 署名・暗号化手順書リスト 1 0 6 に挿入する。

## 【 0 0 1 7 】

図 7 は、図 4 に示した W e b サービス呼出し順 1 0 3 に対して図 6 に示した XML 署名・暗号化手順書取得部 1 0 5 のフローチャートを実行した結果得られる XML 署名・暗号化手順書リスト 1 0 6 の例である。ここで、7 0 6 ~ 7 1 3 はそれぞれ 5 0 8 ~ 5 1 5 に対して対応する W e b サービスの I D を追加すること

により得られた行である。

【0018】

図8はXML署名およびXML暗号化の対象となるXML要素のスキーマ108の例を木構造により表現したものである。ここで、root要素801が文書型であり、801はtickets要素802とhotels要素803とusersinfo要素804を子要素としてもつ。以下同様である。

【0019】

図9はXML署名・暗号化手順解析部107のフローチャート例である。以下、本フローチャートに従って、XML署名・暗号化手順解析時の動作を説明する。まず、変数*i*の値を1に初期化する(901)。次に、108のXMLスキーマを図8に示したように木構造により表したとき、その木を深さ優先で探索し、訪れる各ノードに対して、以下の903から913のステップを行う(902)。まず、訪れたノードのラベル(要素名を表す)を変数*E*に代入する(903)。次に、図7に示したXML署名・暗号化手順書リスト106の中で対象要素703の値が*E*、操作704の値が署名であるような行の集合を求め、それらを*ESL*とする(904)。同様に、106の中で対象要素の値が*E*、操作の値が暗号であるような行の集合を求め、それらを*EEL*とする(905)。次に、*ESL*中の各行*s*について、以下の907から909のステップを行う(906)。まず、*s*のID701の値を*N*、*s*の順序702の値を*S*とする(907)。次に、*EEL*の中にIDの値が*N*で順序の値が*S*より小さいような行が存在するかどうかを判定する(908)。もし存在しなければ、*s*の手順の値を*i*とし、*s*をXML署名・暗号化手順109に挿入した後、*i*の値を1増加させ、*ESL*から*s*を取り除く(909)。次に、*EEL*中の各行*t*について、次の911のステップを行う(910)。911では、*t*の手順の値を*i*とし、*t*を109に挿入した後、*i*の値を1増加させる。最後に、*ESL*中の各行*s*について、次の913のステップを行う(912)。913では、*s*の手順の値を*i*とし、*s*を109に挿入した後、*i*の値を1増加させる。

## 【 0 0 2 0 】

図 1 0 は図 7 に示した XML 署名・暗号化手順書リスト 1 0 6 と図 8 に示した対象要素の XML スキーマ 1 0 8 に対して、図 9 に示した XML 署名・暗号化手順解析部 1 0 7 のフローチャートを実行した結果得られる XML 署名・暗号化手順 1 0 9 の例である。以下、本例における 1 0 7 の動作について説明する。1 0 8 の木を深さ優先で探索したとき、訪れるノードの順は 8 0 5、8 0 6、8 0 2、8 0 7、8 0 3、8 0 8、8 0 9、8 1 0、8 0 4、8 0 1 となる。8 0 5、8 0 6 に対する繰返し実行時では、ステップ 9 0 3 で E の値は t i c k e t となるが、対象要素 7 0 3 の値が t i c k e t であるような行は 1 0 6 には存在しないため、ステップ 9 0 6 からステップ 9 1 3 の処理は実行されない。8 0 2 に対する繰返し実行時では、E の値は t i c k e t s となり、E S L は空集合であるが、E E L には対象要素 7 0 3 の値が t i c k e t s であり操作 7 0 4 の値が暗号である 7 0 6 が含まれる。従って、ステップ 9 1 1 で、7 0 6 の手順が 1 となり、7 0 6 が 1 0 9 に挿入される。8 0 7、8 0 3、8 0 8、8 0 9、8 1 0 に対する繰返し実行時の動作も同様である。

## 【 0 0 2 1 】

次に、8 0 4 に対する繰返し実行時について説明する。このとき、E の値は u s e r i n f o であり、E S L には対象要素 7 0 3 の値が u s e r i n f o であり操作 7 0 4 の値が署名である 7 1 3 が含まれる。E E L には署名対象の値が u s e r i n f o であり操作の値が暗号である 7 0 7 と 7 1 0 が含まれる。ステップ 9 0 7 で、変数 N に 7 1 3 の I D の値である C が代入され、変数 S に 7 1 3 の順序の値である 2 が代入される。次に、ステップ 9 0 8 で、E E L の中に I D の値が C で順序の値が 2 より小さいような行が存在するかどうかの判定が行われる。本例ではそのような行は存在しないため、ステップ 9 0 9 が実行され、7 1 3 の手順が 4 となり、7 1 3 が 1 0 9 に挿入された後、E S L から 7 1 3 が除去される。

## 【 0 0 2 2 】

次に、7 0 7 と 7 1 0 に対してステップ 9 1 1 が実行され、それぞれの手順が 5、6 となり 1 0 9 に挿入される。ステップ 9 1 2 の実行時には E S L は空集合

であるため、913の処理は実行されない。801に対する繰返し実行時の動作は同様であるため省略する。このようにして109が得られる。

#### 【0023】

図11はXML署名・暗号化モジュール出力部110のフローチャート例である。以下、本フローチャートに従ってXML署名・暗号化モジュール出力時の動作を説明する。109の各行tを手順1001の値の昇順で取得し、以下の1102から1107のステップを行う(1101)。まず、tの操作1005の値が署名であるかどうかの判定を行う(1102)。もしそうであれば、tの対象要素1004に対してアルゴリズム1006により署名を行い、署名要素を生成するようなプログラムコードを出力する(1103)。もしそうでなければ、以下の1104から1107のステップを行う。まず、tの対象要素1004をアルゴリズム1006により暗号化を行い、暗号化要素を生成するようなプログラムコードを出力する(1104)。次に、変数Eの値をtの対象要素1004の値、変数Sの値をtの手順1001の値とする(1105)。次に、109に対象要素の値がE、操作の値が暗号、手順の値がSより大きいような行が存在するかどうかを判定する(1106)。もし存在しなければ、対象要素を1104で生成した暗号化要素で置換するようなプログラムコードを出力する(1107)。

#### 【0024】

図12は図10に示したXML署名・暗号化手順109に対し図11に示したXML署名・暗号化モジュール出力部110のフローチャートを実行して得られるXML署名・暗号化モジュール111のフローチャート例である。以下、本フローチャートに従って該XML署名・暗号化モジュールの動作を説明する。まず、XML署名・暗号化の対象要素を含むXML文書を読み込む(1201)。次に、tickets要素をAに対する鍵を使用してAESアルゴリズムにより暗号化して暗号化要素を作成し、tickets要素を作成した暗号化要素で置換する(1202)。次に、hotels要素をBに対する鍵を使用してDESe deアルゴリズムにより暗号化して暗号化要素を作成し、hotels要素を作成した暗号化要素で置換する(1203)。次に、cardinfo要素をCの

鍵を使用してRSAアルゴリズムにより暗号化して暗号化要素を作成し、cardinfo要素を作成した暗号化要素で置換する(1204)。次に、userinfo要素に対してDSSアルゴリズムにより署名を行い、署名要素を作成する(1205)。次に、userinfo要素をAに対する鍵でDESedeアルゴリズムにより暗号化して暗号化要素を作成する(1206)。次に、userinfo要素をBに対する鍵でAESアルゴリズムにより暗号化して暗号化要素を作成し、userinfo要素を作成した暗号化要素で置換する(1207)。次に、root要素に対してDSSアルゴリズムにより署名を行い、署名要素を作成する(1208)。最後に、root要素に対してRSAアルゴリズムにより署名を行い、署名要素を作成する(1209)。

## 【0025】

図13はXML署名・暗号化モジュール登録部112のフローチャート例である。以下、本フローチャートに従ってXML署名・暗号化モジュール登録時の動作を説明する。まず、108のXMLスキーマのスキーマURIをS、103のWebサービス呼出し順の識別子であるパスURIをP、110で作成したXML署名・暗号化モジュールのIDをMとする(1301)。次に、S、P、Mからなる組をXML署名・暗号化モジュール対応表113に挿入する(1302)。

## 【0026】

図14はXML署名・暗号化モジュール対応表113の例である。110で新たなXML署名・暗号化モジュールが生成される毎に、113に該モジュールに対応する行が挿入される。

## 【0027】

図15はXML署名・暗号化実行部114のフローチャート例である。以下、本フローチャートに従ってXML署名・暗号化実行時の動作を説明する。まず、XML署名・暗号化の対象要素を含むXML文書Dを受け取る(1501)。次に、D内の記述からスキーマURIを表すSとパスURIを表すPを取得する(1502)。次に、XML署名・暗号化モジュール対応表113からスキーマURI1401の値がSであり、パスURI1402の値がPであるような行を検

索し、そのモジュールID 1 4 0 3 の値から対応するXML署名・暗号化モジュールを決定する(1 5 0 3)。次に、Dに対して該XML署名・暗号化モジュールを実行する(1 5 0 4)。最後に、該XML署名・暗号化モジュールの実行結果をWebサービスの送信文書として送信する(1 5 0 5)。

## 【0 0 2 8】

図1 6はXML文書1 1 5の例である。ここで、path要素はWebサービスの呼出し順を表す要素であり、root要素はXMLスキーマ1 0 8に適合する形式をもつ要素である。path要素のxmlns属性の値がパスURIを表し、root要素のxmlns属性の値がスキーマURIを表す。

## 【0 0 2 9】

図1 7は図1 6に示したXML文書1 1 5に対して図1 5に示したXML署名・暗号化実行部1 1 4のフローチャートを実行して得られるWebサービス送信文書1 1 6の例である。記述は一部省略する。ここで、Signature要素、EncryptedData要素はそれぞれXML署名・暗号化モジュール1 1 1により生成された署名要素、暗号化要素である。本例では、1 1 4により実行されるXML署名・暗号化モジュールのIDは、1 1 5に示されたスキーマURIの値とパスURIの値が一致する行1 4 0 4のモジュールID 1 4 0 3の値であるXMLSEC0 1である。

## 【0 0 3 0】

次に、図1 8を用いて本発明の第2の実施形態によるXML署名およびXML暗号化方法について説明する。第1の実施形態では、開発時にあらかじめXML署名・暗号化モジュールを作成しておき、実行時に該モジュールを呼び出す方式であるのに対し、第2の実施形態では、実行時にXML署名・暗号化手順の解析とXML署名・暗号化モジュールの生成を行う。

## 【0 0 3 1】

図1 8は本実施形態によるXML署名およびXML暗号化方法の概要である。XML署名・暗号化処理系1 8 0 1はXML署名・暗号化手順書取得部1 8 0 2とXML署名・暗号化手順解析部1 0 7とXML署名・暗号化モジュール出力部1 1 0とXML署名・暗号化実行部1 8 0 3からなる。



1801はXML文書115を入力として呼び出され、まず1802が実行される。1802は該XML文書に記述されたWebサービスURIをもとに各WebサービスのXML署名・暗号化手順書を取得する。XML署名・暗号化手順書リスト106の作成方法は105と同様である。その後、XML署名・暗号化モジュール111の生成までの処理は前記第1の実施形態と同様である。

【0032】

XML署名・暗号化実行部1803は115に対して110により生成されたXML署名・暗号化モジュール111を実行し、その結果得られるXML文書116をWebサービスの送信文書として送信する。

【0033】

【発明の効果】

本発明によれば、複数のWebサービスを利用するようなWebサービスを新規に開発するときに、利用するWebサービスが要求するすべてのXML署名・暗号化手順を満たすような手順でXML署名・暗号化を行うモジュールが自動生成されるため、開発者の負担を軽減することができる。

【図面の簡単な説明】

【図1】 本発明の一実施形態のシステム構成概要を示す図。

【図2】 システム構成のハードウェア構成図。

【図3】 Webサービス呼出し順定義画面102の例を示す図。

【図4】 Webサービス呼出し順103の例を示す図。

【図5】 XML署名・暗号化手順書104の例を示す図。

【図6】 XML署名・暗号化手順書取得部105のフローチャート。

【図7】 XML署名・暗号化手順書リスト106の例を示す図。

【図8】 対象要素のXMLスキーマ108の例を示す図。

【図9】 XML署名・暗号化手順解析部107のフローチャート。

【図10】 XML署名・暗号化手順109の例を示す図。

【図11】 XML署名・暗号化モジュール出力部110のフローチャート。

【図12】 XML署名・暗号化モジュール111のフローチャート。

【図13】 XML署名・暗号化モジュール登録部112のフローチャート。

【図 1 4】XML 署名・暗号化モジュール対応表 1 1 3 の例を示す図。

【図 1 5】XML 署名・暗号化実行部 1 1 4 のフローチャート。

【図 1 6】XML 文書 1 1 5 の例を示す図。

【図 1 7】Web サービス送信文書 1 1 6 の例を示す図。

【図 1 8】本発明の第二の実施形態によるシステム構成を示すブロック図。

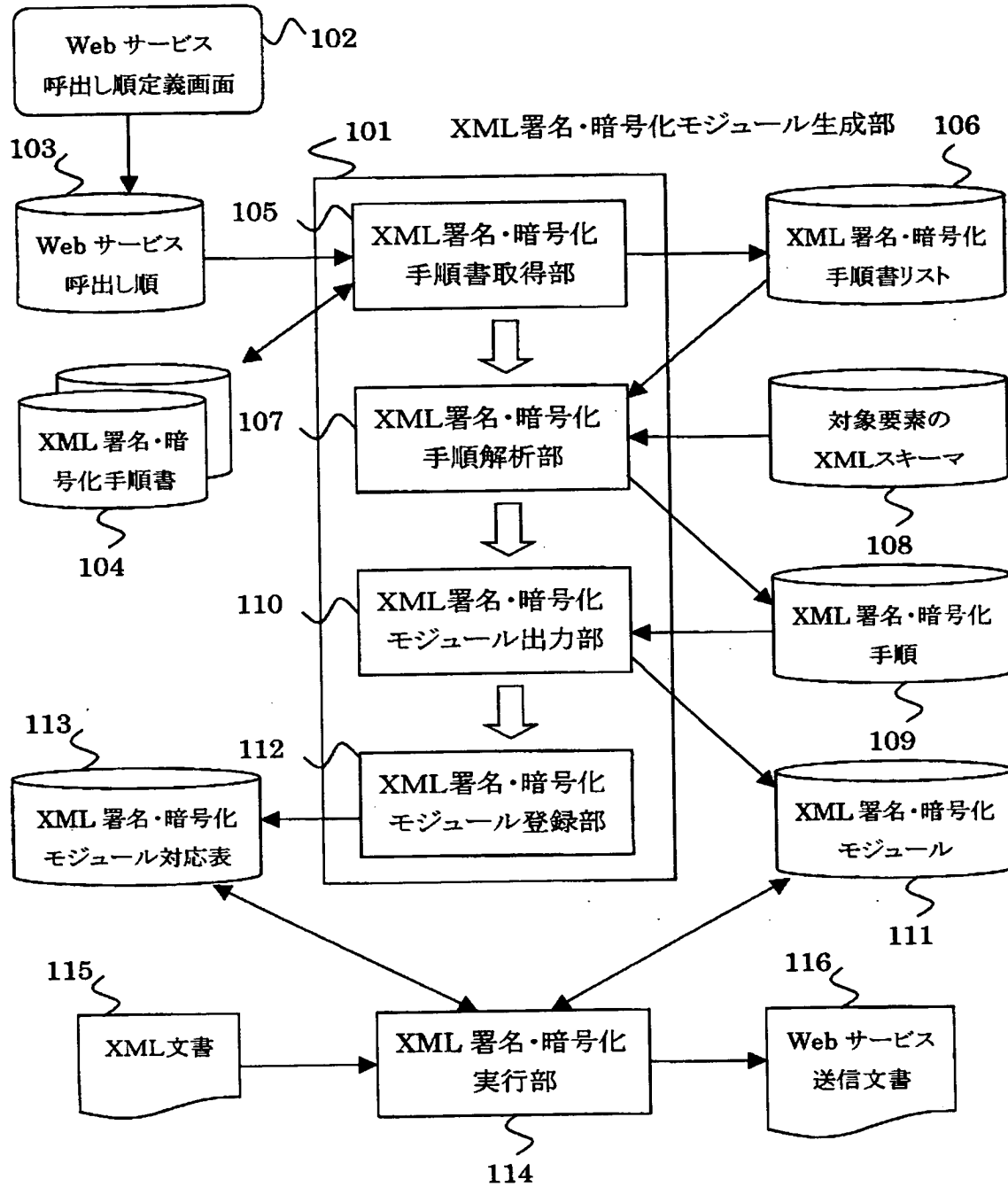
【符号の説明】

1 0 1 …XML 署名・暗号化モジュール生成部、1 0 2 …Web サービス呼出し  
順定義画面、1 0 3 …Web サービス呼出し順、1 0 4 …XML 署名・暗号化手  
順書、1 0 5 …XML 署名・暗号化手順書取得部、1 0 6 …XML 署名・暗号化  
手順書リスト、1 0 7 …XML 署名・暗号化手順解析部、1 0 8 …対象要素の X  
ML スキーマ、1 0 9 …XML 署名・暗号化手順、1 1 0 …XML 署名・暗号化  
モジュール出力部、1 1 1 …XML 署名・暗号化モジュール、1 1 2 …XML 署  
名・暗号化モジュール登録部、1 1 3 …XML 署名・暗号化モジュール対応表、  
1 1 4 …XML 署名・暗号化実行部、1 1 5 …XML 文書、1 1 6 …Web サー  
ビス送信文書、1 8 0 1 …XML 署名・暗号化処理系。

【書類名】 図面

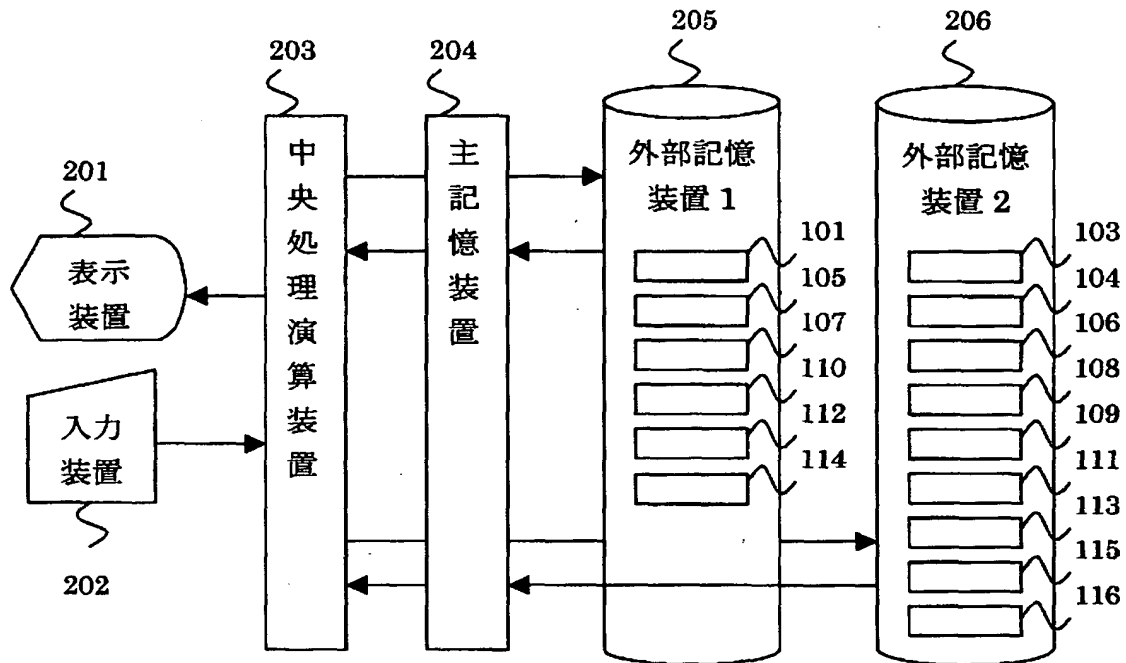
【図 1】

図 1



【図 2】

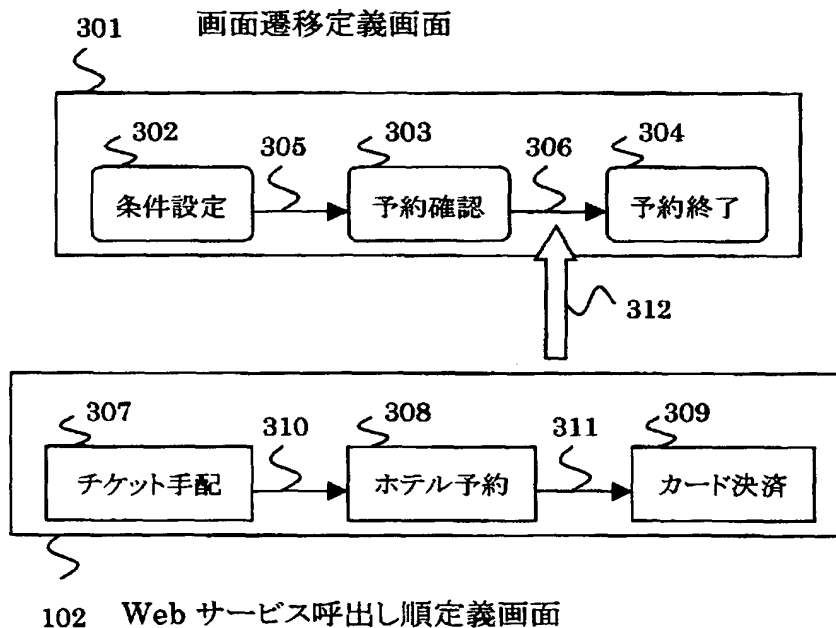
図 2



【図 3】

図 3

Web サービス呼出し順定義画面 102



【図 4】

図 4

Web サービス呼出し順 103

	401	402	403	404	405
	Σ	Σ	Σ	Σ	Σ
	ID	次 ID	名称	Web サービス URI	XML 署名・暗号化手順書 URI
406 ✓	A	B	チケット手配	<a href="http://www.tickets.com/">http://www.tickets.com/</a>	<a href="http://www.tickets.com/sec">http://www.tickets.com/sec</a>
407 ✓	B	C	ホテル予約	<a href="http://www.hotels.com/">http://www.hotels.com/</a>	<a href="http://www.hotels.com/sec">http://www.hotels.com/sec</a>
408 ✓	C	なし	カード決済	<a href="http://www.cards.com/">http://www.cards.com/</a>	<a href="http://www.cards.com/sec">http://www.cards.com/sec</a>

【図 5】

図 5

XML 署名・暗号化手順書 104

406 の XML 署名・暗号化手順書 501

	504	505	506	507
	⌋	⌋	⌋	⌋
508	順序	対象要素	操作	アルゴリズム
509	1	tickets	暗号	AES
510	2	userinfo	暗号	DESede
	3	root	署名	DSS

407 の XML 署名・暗号化手順書 502

511	順序	対象要素	操作	アルゴリズム
512	1	hotels	暗号	DESede
513	2	userinfo	暗号	AES
	3	root	署名	RSA

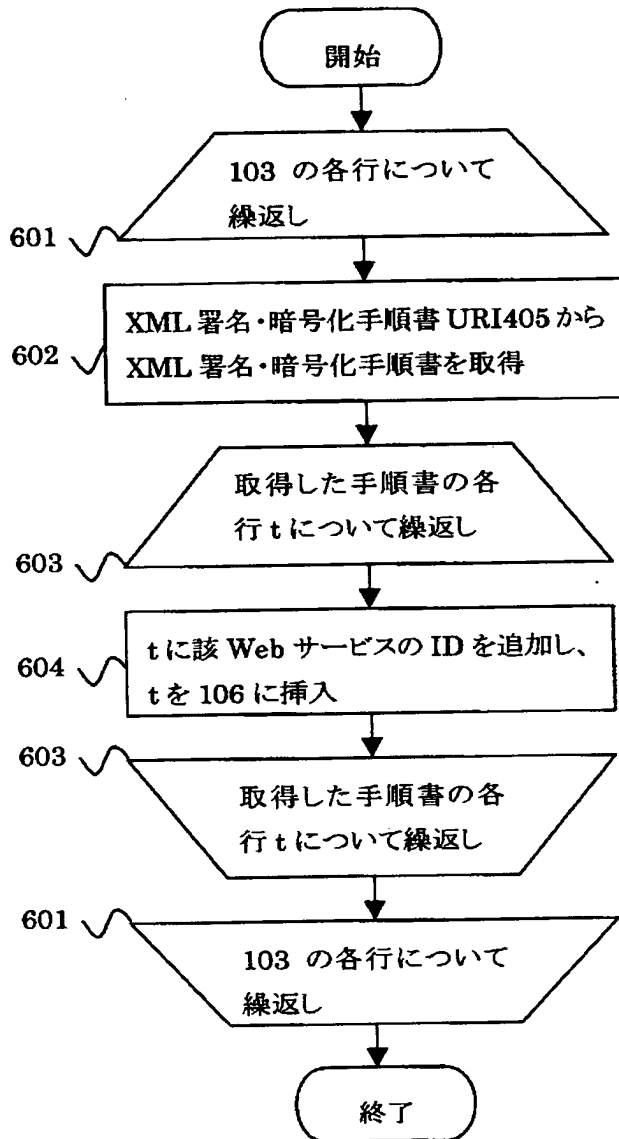
408 の XML 署名・暗号化手順書 503

514	順序	対象要素	操作	アルゴリズム
515	1	cardinfo	暗号	RSA
	2	userinfo	署名	DSS

【図 6】

図 6

XML 署名・暗号化手順書取得部 105



【図 7】

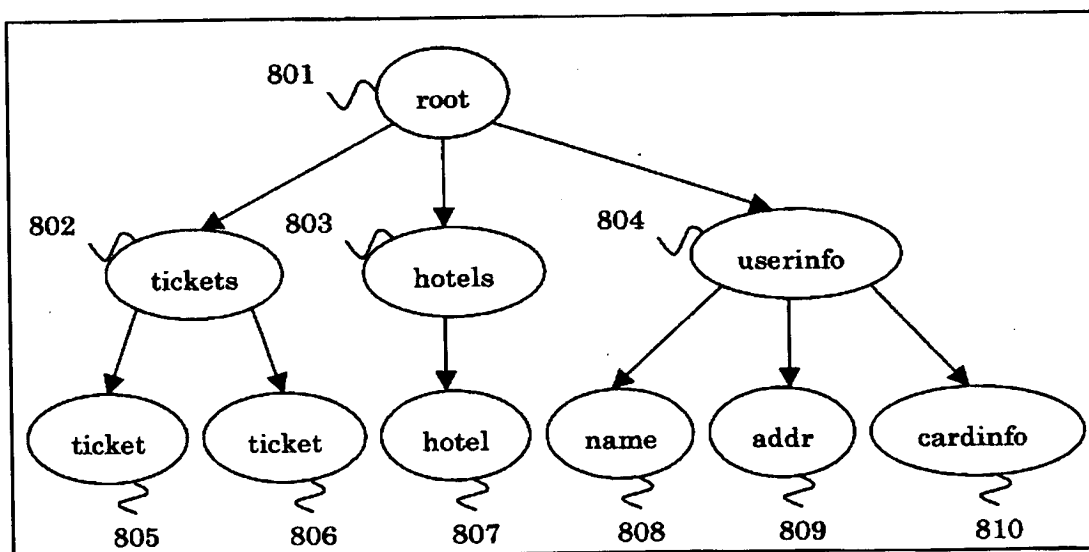
図 7

XML 署名・暗号化手順書リスト 106

	701	702	703	704	705
	ID	順序	対象要素	操作	アルゴリズム
706	A	1	tickets	暗号	AES
707	A	2	userinfo	暗号	DESede
708	A	3	root	署名	DSS
709	B	1	hotels	暗号	DESede
710	B	2	userinfo	暗号	AES
711	B	3	root	署名	RSA
712	C	1	cardinfo	暗号	RSA
713	C	2	userinfo	署名	DSS

【図 8】

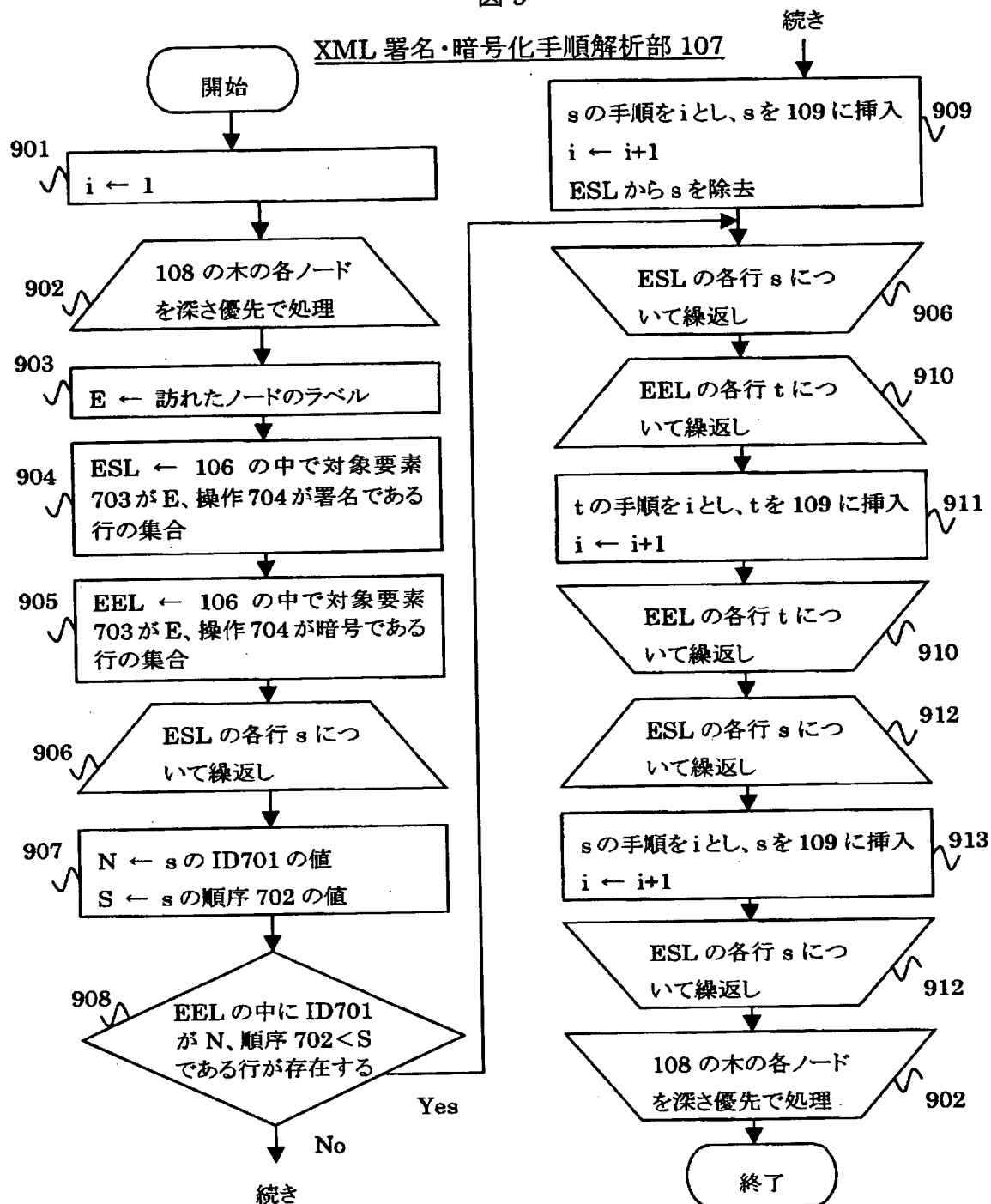
図 8

対象要素の XML スキーマ 108



【図 9】

図 9



【図 1 0】

図 10

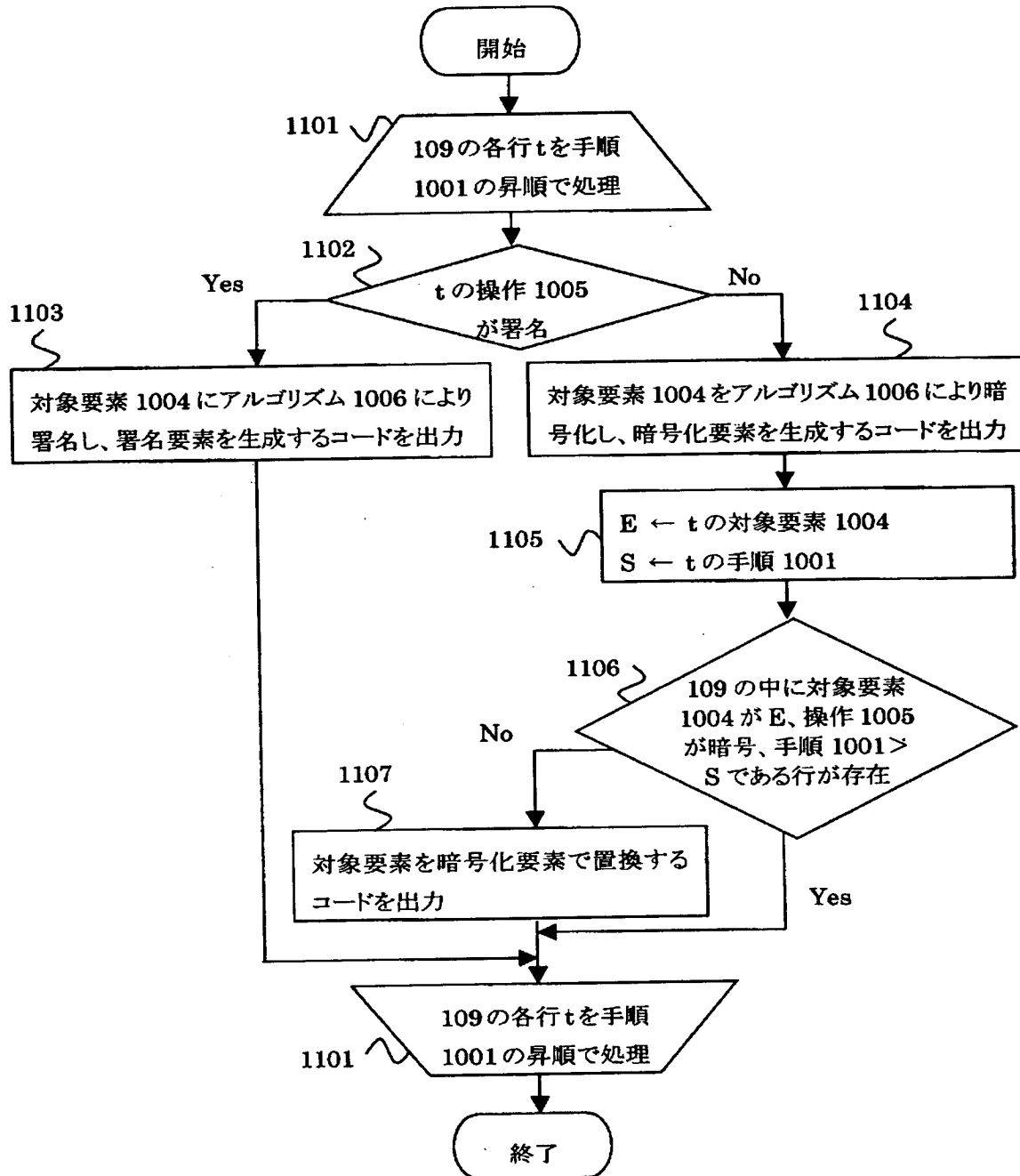
XML 署名・暗号化手順 109

	1001	1002	1003	1004	1005	1006
	手順	ID	順序	対象要素	操作	アルゴリズム
1007 ✓	1	A	1	tickets	暗号	AES
1008 ✓	2	B	1	hotels	暗号	DESede
1009 ✓	3	C	1	cardinfo	暗号	RSA
1010 ✓	4	C	2	userinfo	署名	DSS
1011 ✓	5	A	2	userinfo	暗号	DESede
1012 ✓	6	B	2	userinfo	暗号	AES
1013 ✓	7	A	3	root	署名	DSS
1014 ✓	8	B	3	root	署名	RSA

【図 11】

図 11

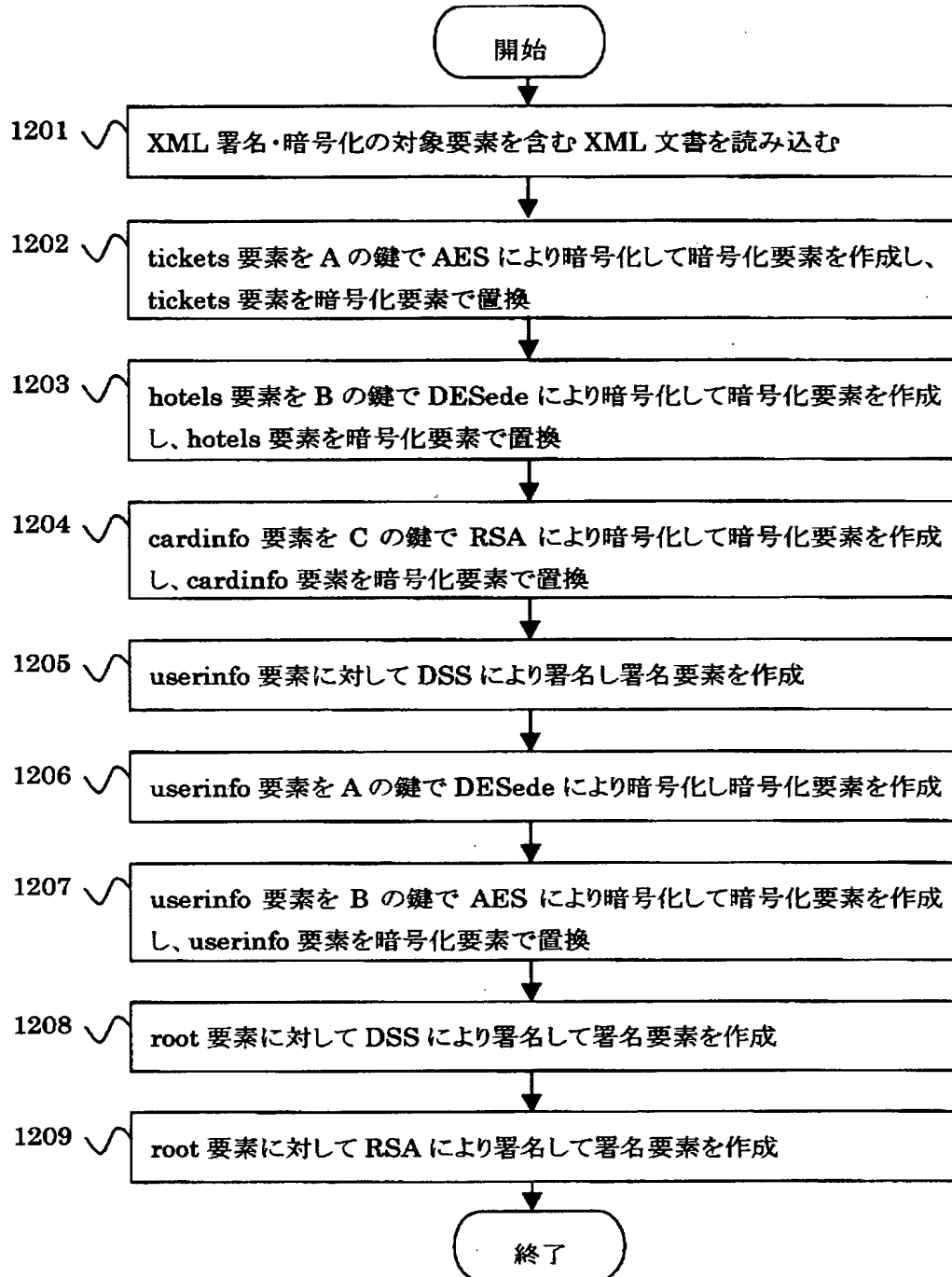
XML 署名・暗号化モジュール出力部 110



【図 1 2】

図 12

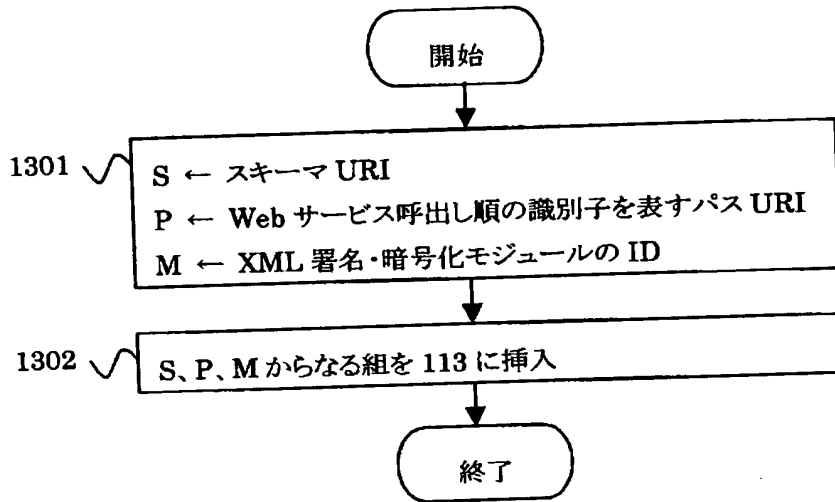
XML 署名・暗号化モジュール 111



【図 13】

図 13

XML 署名・暗号化モジュール登録部 112



【図 14】

図 14

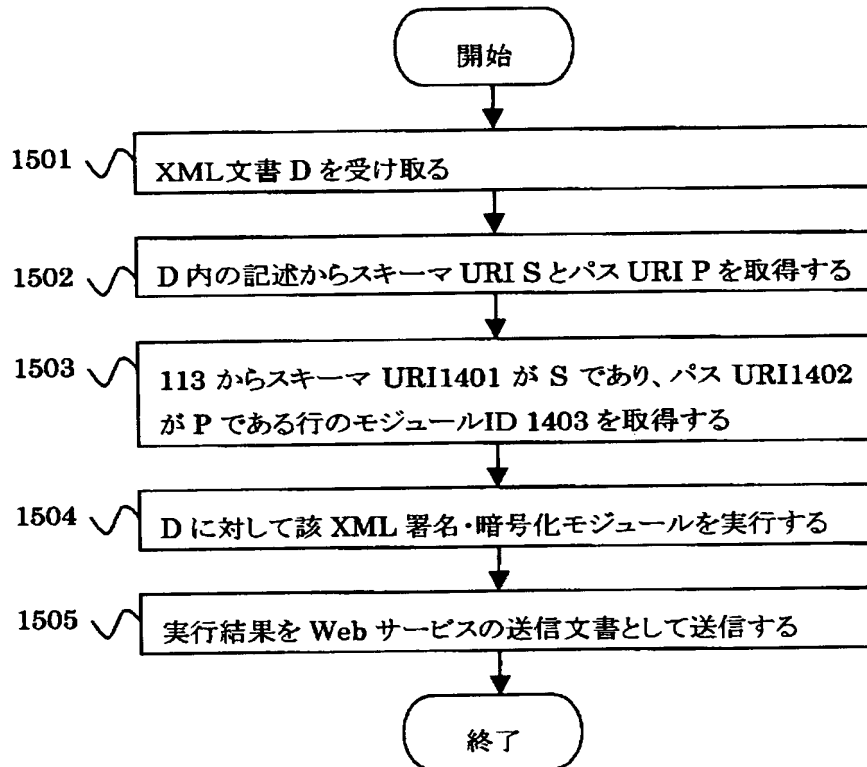
XML 署名・暗号化モジュール対応表 113

	1401 スキーマ URI	1402 パス URI	1403 モジュール ID
1404	http://www.hitachi.co.jp/travel	http://www.hitachi.co.jp/tp1	XMLSEC01
1405	http://www.hitachi.co.jp/travel	http://www.hitachi.co.jp/tp2	XMLSEC02
1406	http://www.hitachi.co.jp/travel	http://www.hitachi.co.jp/bp	XMLSEC03
1407	http://www.hitachi.co.jp/travel	http://www.hitachi.co.jp/pp	XMLSEC04

【図 1 5】

図 15

XML 署名・暗号化実行部 114



【図 1 6】

図 16

XML 文書 115

```
<?xml version="1.0" encoding="Shift_JIS"?>
<message>
  <path xmlns="http://www.hitachi.co.jp/tp1">
    <next URI="http://www.tickets.com/" />
    <next URI="http://www.hotels.com/" />
    <next URI="http://www.cards.com/" />
  </path>
  <root xmlns="http://www.hitachi.co.jp/travel">
    <tickets>
      <ticket from="東京" to="大阪" date="02/04/01" time="09:00" />
      <ticket from="大阪" to="東京" date="02/04/02" time="17:00" />
    </tickets>
    <hotels>
      <hotel name="日立ホテル" roomtype="S" date="02/04/01" />
    </hotels>
    <userinfo>
      <name>日立太郎</name>
      <addr>東京都千代田区</addr>
      <cardinfo expiration="04/04" cardnumber="0123 4567 8901 2345" />
    </userinfo>
  </root>
</message>
```

【図 17】

図 17

## Web サービス送信文書 116

```

<?xml version="1.0" encoding="Shift_JIS"?>
<message>
  <path xmlns="http://www.hitachi.co.jp/tp1">
    <next URI="http://www.tickets.com/" />
    <next URI="http://www.hotels.com/" />
    <next URI="http://www.cards.com/" />
  </path>
  <root xmlns="http://www.hitachi.co.jp/travel">
    <EncryptedData xmlns="http://.../xmlenc#" Recipient="チケット手配">
      <CipherData><CipherValue>AB...CD</CipherValue></CipherData>
    </EncryptedData>
    <EncryptedData xmlns="http://.../xmlenc#" Recipient="ホテル予約">
      <CipherData><CipherValue>EF...GH</CipherValue></CipherData>
    </EncryptedData>
    <EncryptedData xmlns="http://.../xmlenc#" Recipient="チケット手配">
      <CipherData><CipherValue>HI...JK</CipherValue></CipherData>
    </EncryptedData>
    <EncryptedData xmlns="http://.../xmlenc#" Recipient="ホテル予約">
      <CipherData><CipherValue>LM...NO</CipherValue></CipherData>
    </EncryptedData>
  </root>
  <Signature xmlns="http://.../xmldsig#" Id="チケット手配">
    <SignatureValue>ABC...DEF</SignatureValue>
  </Signature>
  <Signature xmlns="http://.../xmldsig#" Id="ホテル予約">
    <SignatureValue>BCD...EFG</SignatureValue>
  </Signature>
  <Signature xmlns="http://.../xmldsig#" Id="カード決済">
    <SignatureValue>CDE...FGH</SignatureValue>
  </Signature>
</message>

```



【書類名】 要約書

【要約】

【解決手段】

XML署名・暗号化手順リスト106と対象要素のXMLスキーマ108を入力としてXML署名・暗号化手順解析部107を実行することにより、要求されるすべての手順を満たすようなXML署名・暗号化手順109を出力する。XML署名・暗号化モジュール出力部110は109を参照し、XML署名・暗号化を行うモジュール111を生成する。

【効果】

利用するWebサービスが要求するすべてのXML署名・暗号化手順を満たすようにXML署名・暗号化を行うXML署名・暗号化モジュールが自動生成されるため、開発者の負担を軽減することができる。

【選択図】 図1

認定・付加情報

特許出願の番号	特願 2 0 0 3 - 0 4 6 6 8 3	
受付番号	5 0 3 0 0 2 9 6 3 8 1	
書類名	特許願	
担当官	第八担当上席	0 0 9 7
作成日	平成 1 5 年	2 月 2 6 日

< 認定情報・付加情報 >

【提出日】	平成15年 2月25日
-------	-------------

出 願 人 履 歴 情 報

識別番号 [000005108]

1. 変更年月日 1990年 8月31日

[変更理由] 新規登録

住 所 東京都千代田区神田駿河台4丁目6番地  
氏 名 株式会社日立製作所